

本节内容

# 线索二叉树

找前驱/后继

## 知识总览

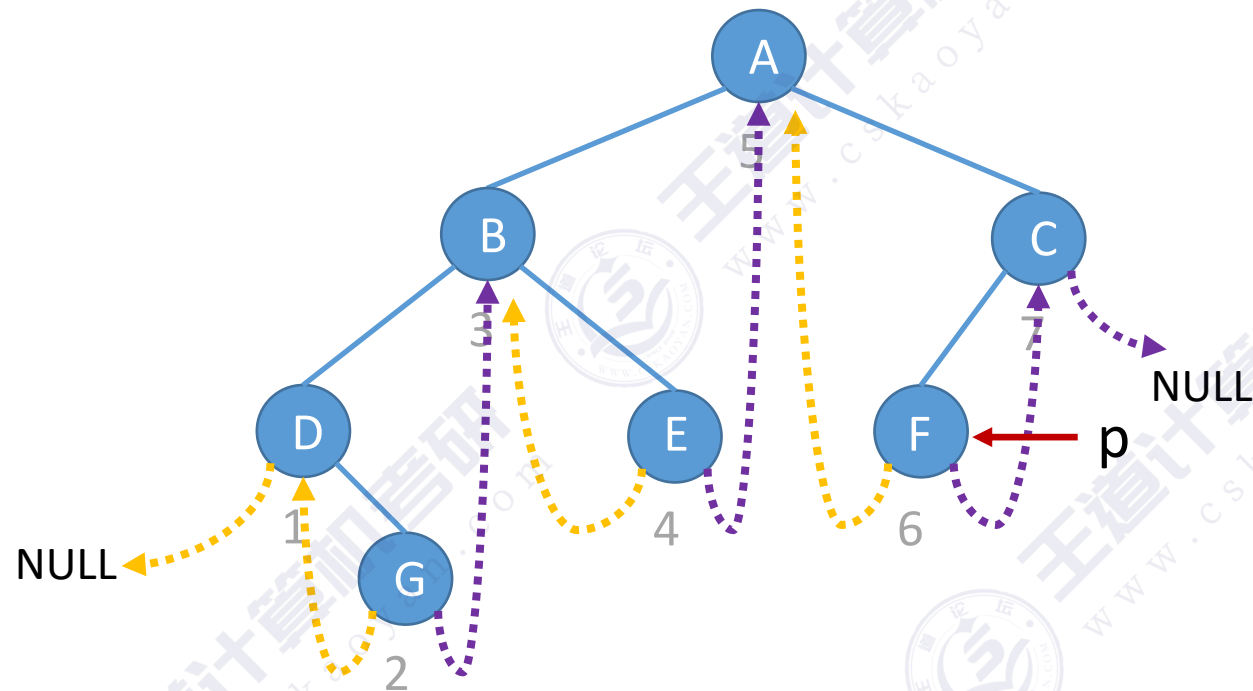
### 线索二叉树找前驱/后继

中序线索二叉树

先序线索二叉树

后序线索二叉树

## 中序线索二叉树找中序后继

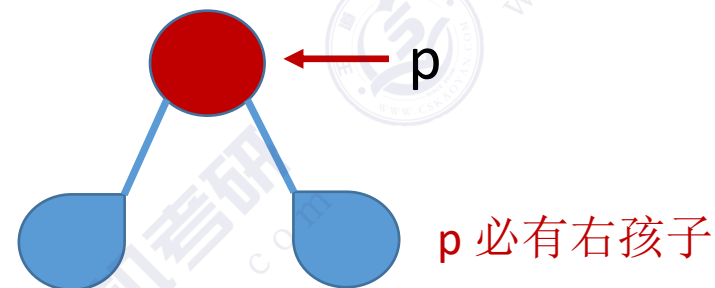


中序遍历序列: D G B E A F C

在中序线索二叉树中找到指定结点\*p  
的中序后继 next

①若  $p \rightarrow rtag == 1$ , 则  $next = p \rightarrow rchild$

②若  $p \rightarrow rtag == 0$



中序遍历——左 根 右

左 根 (左 根 右)

左 根 ((左 根 右) 根 右)

$next = p$  的右子树中最左下结点

## 中序线索二叉树找中序后继

//找到以P为根的子树中，第一个被中序遍历的结点

```
ThreadNode *Firstnode(ThreadNode *p){  
    //循环找到最左下结点(不一定是叶结点)  
    while(p->rtag==0) p=p->lchild;  
    return p;  
}
```

//在中序线索二叉树中找到结点p的后继结点

```
ThreadNode *Nextnode(ThreadNode *p){  
    //右子树中最左下结点  
    if(p->rtag==0) return Firstnode(p->rchild);  
    else return p->rchild; //rtag==1直接返回后继线索  
}
```

//对中序线索二叉树进行中序遍历（利用线索实现的非递归算法）

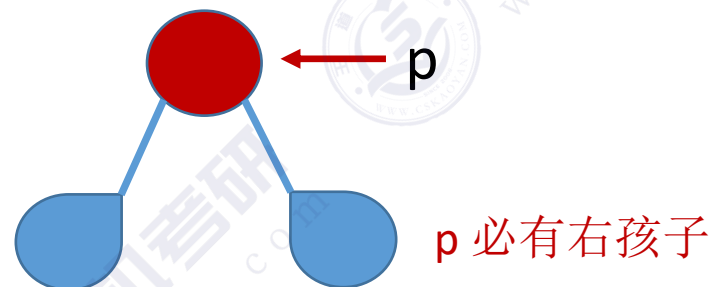
```
void Inorder(ThreadNode *T){  
    for(ThreadNode *p=Firstnode(T);p!=NULL;p=Nextnode(p))  
        visit(p);  
}
```

空间复杂度O(1)

在中序线索二叉树中找到指定结点\*p的中序后继 next

①若  $p \rightarrow rtag == 1$ ，则  $next = p \rightarrow rchild$

②若  $p \rightarrow rtag == 0$



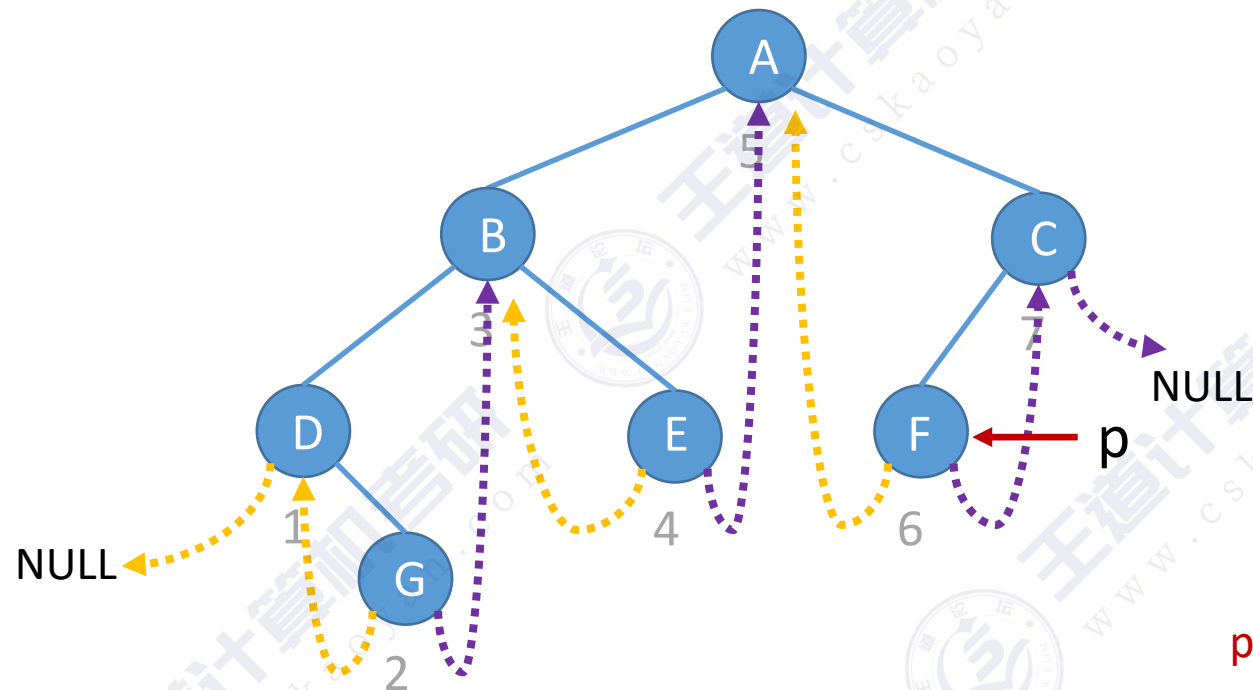
中序遍历——左 根 右

左 根 (左 根 右)

左 根 ((左 根 右) 根 右)

next = p的右子树中最左下结点

## 中序线索二叉树找中序前驱



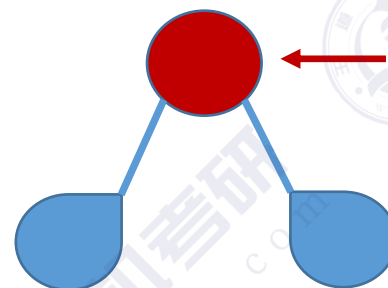
中序遍历序列: D G B E A F C

在中序线索二叉树中找到指定结点\*p  
的中序前驱 pre

①若  $p \rightarrow ltag == 1$ , 则  $pre = p \rightarrow lchild$

②若  $p \rightarrow ltag == 0$

p 必有左孩子



中序遍历——左 根 右

(左 根 右) 根 右

(左 根 (左 根 右)) 根 右

pre = p 的左子树中最右下结点



## 中序线索二叉树找中序前驱

//找到以P为根的子树中，最后一个被中序遍历的结点

```
ThreadNode *Lastnode(ThreadNode *p){  
    //循环找到最右下结点(不一定是叶结点)  
    while(p->rtag==0) p=p->rchild;  
    return p;  
}
```

//在中序线索二叉树中找到结点p的前驱结点

```
ThreadNode *Prenode(ThreadNode *p){  
    //左子树中最右下结点  
    if(p->ltag==0) return Lastnode(p->lchild);  
    else return p->lchild; //ltag==1直接返回前驱线索  
}
```

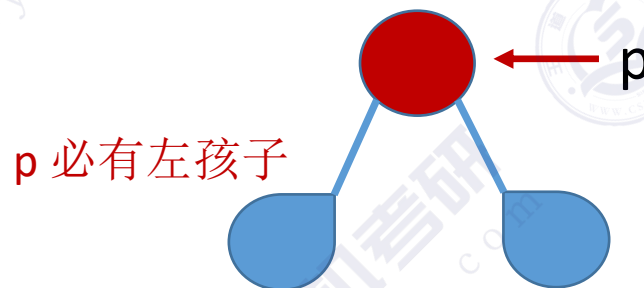
//对中序线索二叉树进行逆向中序遍历

```
void RevInorder(ThreadNode *T){  
    for(ThreadNode *p=Lastnode(T);p!=NULL;p=Prenode(p))  
        visit(p);  
}
```

在中序线索二叉树中找到指定结点\*p  
的中序前驱 pre

①若  $p \rightarrow ltag == 1$ ，则  $pre = p \rightarrow lchild$

②若  $p \rightarrow ltag == 0$



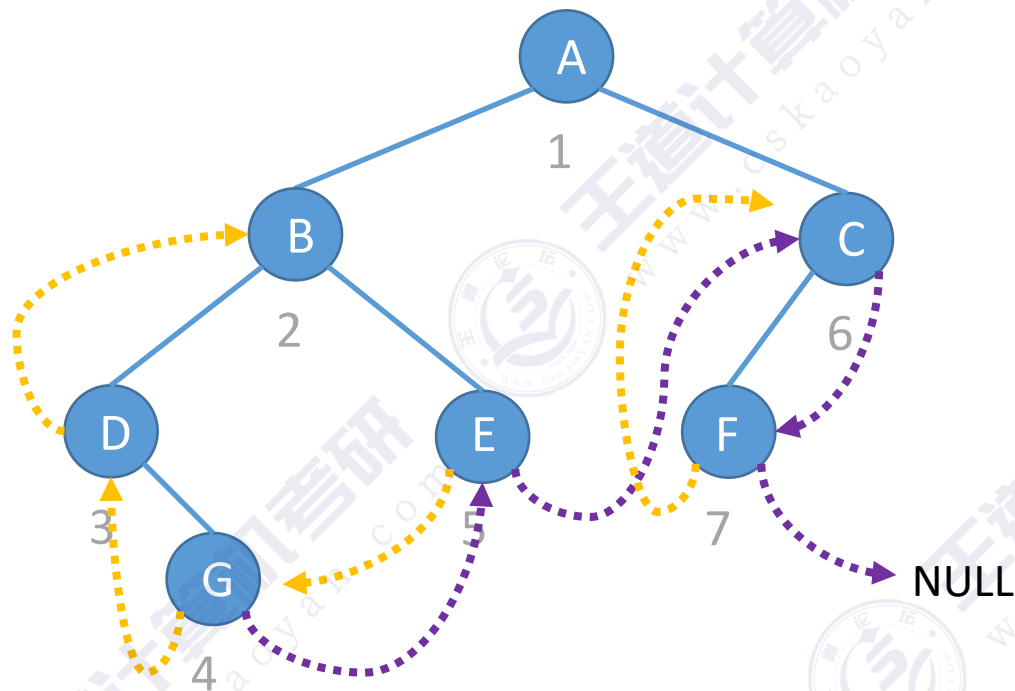
中序遍历——左 根 右

(左 根 右) 根 右

(左 根 (左 根 右)) 根 右

pre = p的左子树中最右下结点

# 先序线索二叉树找先序后继



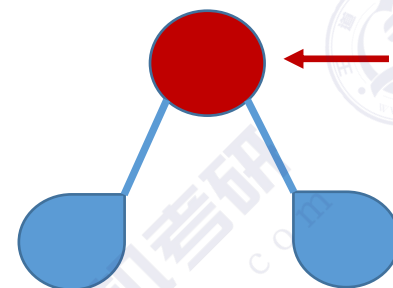
先序遍历序列: A B D G E C F



在先序线索二叉树中找到指定结点\*p  
的先序后继 next

①若  $p \rightarrow rtag == 1$ , 则  $next = p \rightarrow rchild$

②若  $p \rightarrow rtag == 0$



p 必有右孩子

若p有左孩子,  
则先序后继为  
左孩子

先序遍历——根 左 右  
根 (根 左 右) 右

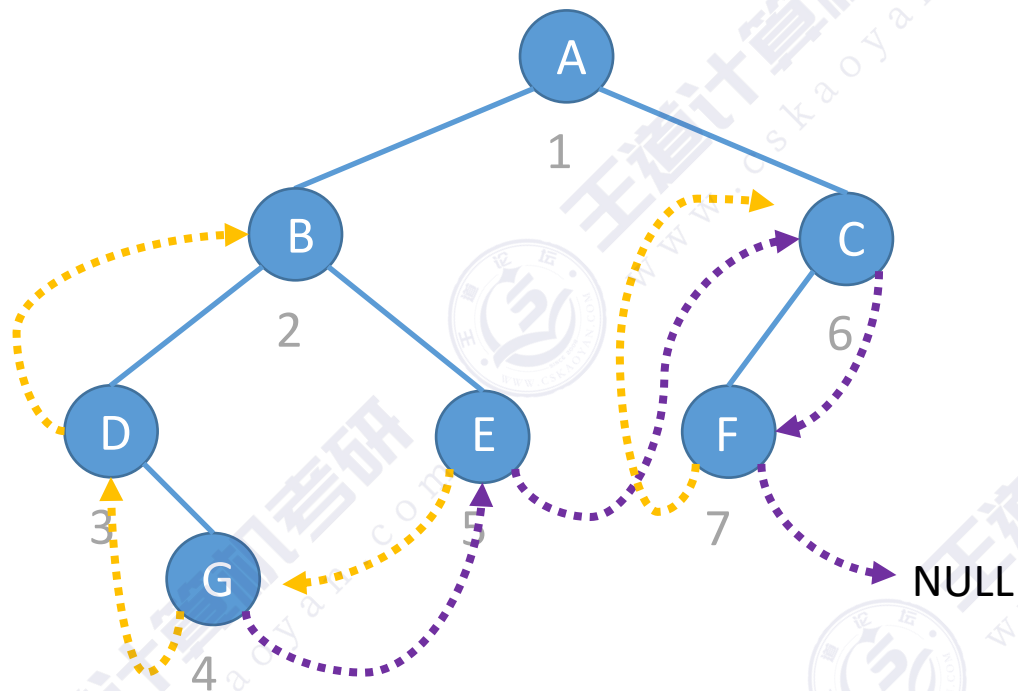
假设有左孩子

若p没有左孩  
子, 则先序后  
继为右孩子

先序遍历——根 右  
根 (根 左 右)

假设没有左孩子

# 先序线索二叉树找先序前驱



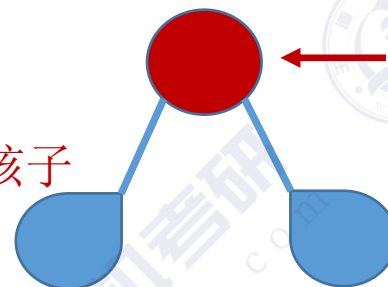
先序遍历序列: A B D G E C F

在先序线索二叉树中找到指定结点\*p  
的先序前驱 pre

①若  $p \rightarrow ltag == 1$ , 则  $next = p \rightarrow lchild$

②若  $p \rightarrow ltag == 0$

p 必有左孩子



先序遍历——根 左 右



荒唐的答案

除非用土办法  
从头开始先序  
遍历

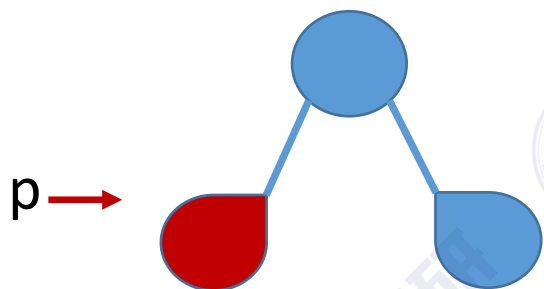
先序遍历中, 左右子树  
中的结点只可能是根  
的后继, 不可能是前驱



改用三叉链表可以找到父节点

## 先序线索二叉树找先序前驱

①如果能找到  $p$  的父节点，且  $p$  是左孩子

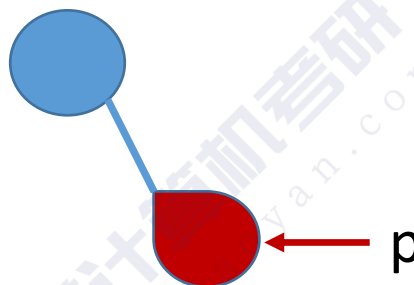


先序遍历——根 左 右

根 (根 左 右) 右

$p$  的父节点即为其前驱

②如果能找到  $p$  的父节点，且  $p$  是右孩子，其左兄弟为空

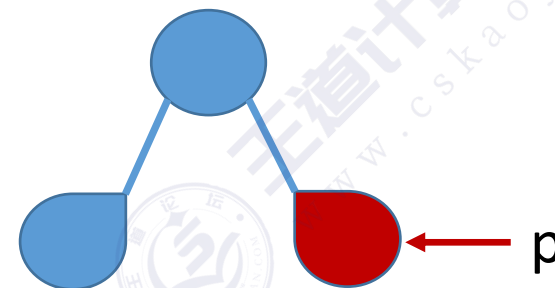


先序遍历——根 右

根 (根 左 右)

$p$  的父节点即为其前驱

③如果能找到  $p$  的父节点，且  $p$  是右孩子，其左兄弟非空

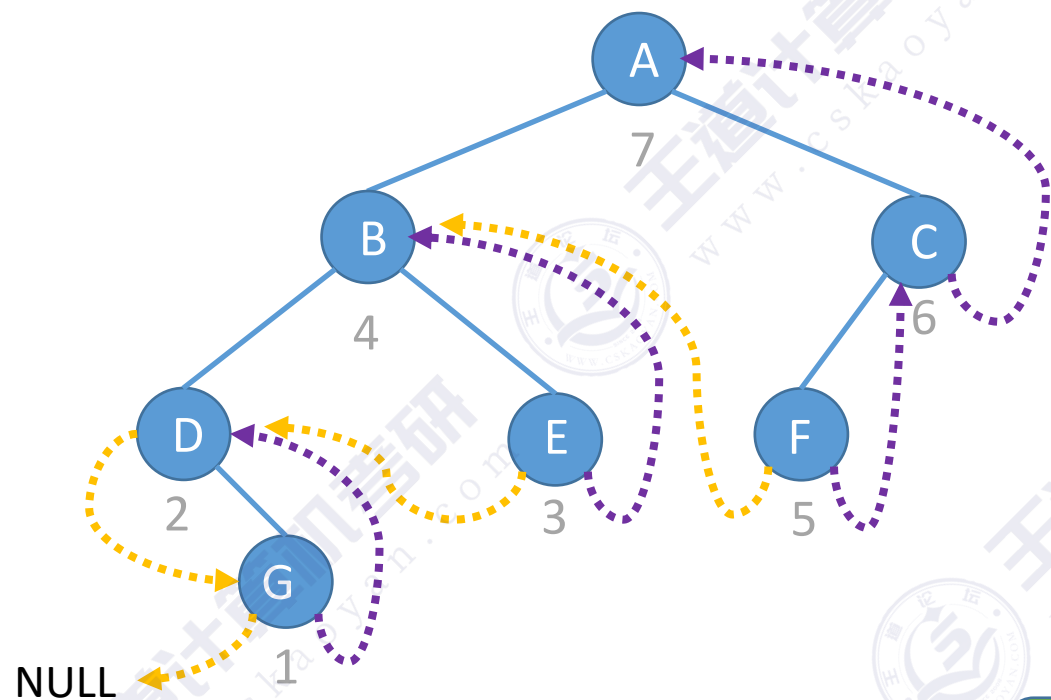


根 左 右

$p$  的前驱为左兄弟子树中最后一个被先序遍历的结点

④如果  $p$  是根节点，则  $p$  没有先序前驱

## 后序线索二叉树找后序前驱



后序遍历序列: G D E B F C A

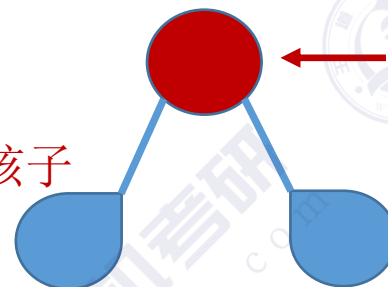


在后序线索二叉树中找到指定结点\*p  
的**后序前驱** pre

①若  $p \rightarrow ltag == 1$ , 则  $pre = p \rightarrow lchild$

②若  $p \rightarrow ltag == 0$

p 必有左孩子



若p有右孩子,  
则后序前驱为  
右孩子

后序遍历——左 右 根

假设有右孩子

左 (左 右 根) 根

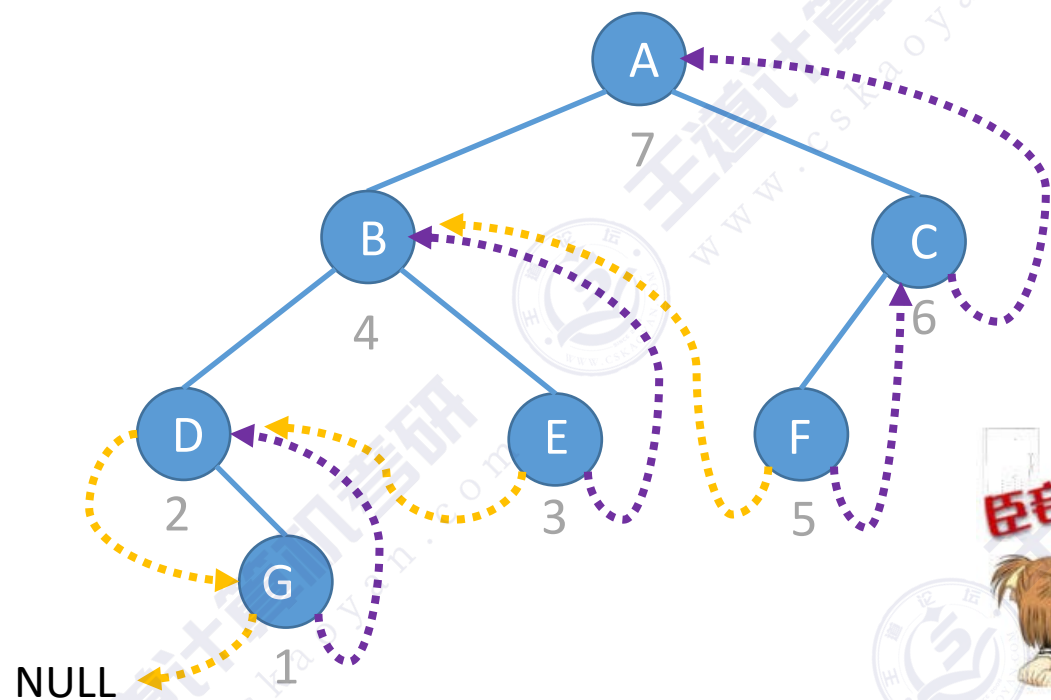
若p没有右孩  
子, 则后序前  
驱为左孩子

后序遍历——左 根

假设没有右孩子

(左 右 根) 根

# 后序线索二叉树找后序后继

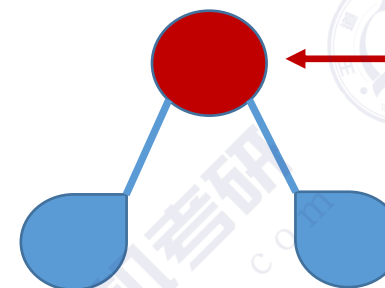


后序遍历序列: G D E B F C A

在后序线索二叉树中找到指定结点\*p  
的**后序后继** next

①若  $p \rightarrow rtag == 1$ , 则  $next = p \rightarrow rchild$

②若  $p \rightarrow rtag == 0$



p 必有右孩子

后序遍历——左 右 根



荒唐的答案

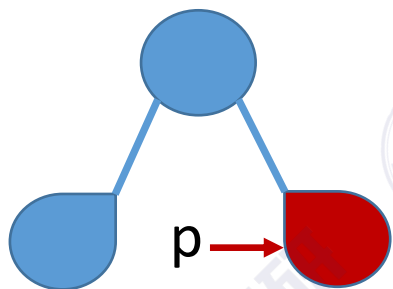
除非用土办法  
从头开始先序  
遍历

后序遍历中, 左右子树  
中的结点只可能是根的  
前驱, 不可能是后继

改用三叉链表可以  
找到父节点

## 后序线索二叉树找后序后继

①如果能找到  $p$  的父节点，  
且  $p$  是右孩子

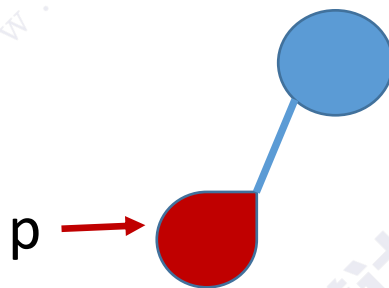


后序遍历——左 右 根

左 (左 右 根) 根

$p$  的父节点即为其后继

②如果能找到  $p$  的父节点，且  
 $p$  是左孩子，其右兄弟为空

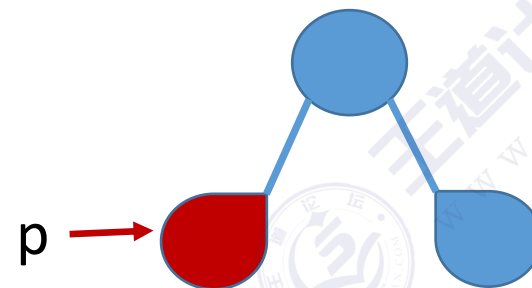


后序遍历——左 根

(左 右 根) 根

$p$  的父节点即为其后继

③如果能找到  $p$  的父节点，且  
 $p$  是左孩子，其右兄弟非空



左 右 根

$p$  的后继为右兄弟子树中  
第一个被后序遍历的结点

④如果  $p$  是根节点，则  $p$  没有后序后继



# 知识回顾与重要考点

若 ltag/rtag == 0

中序线索  
二叉树

后继

p的右子树中最左下结点

前驱

p的左子树中最右下结点

先序线索  
二叉树

后继

①若p有左孩子，则先序后继为左孩子

②若p没有左孩子，则先序后继为右孩子

前驱

①如果能找到 p 的父节点，且p是左孩子，则p的父节点为其前驱

②如果能找到 p 的父节点，且p是右孩子，其左兄弟为空，则p的父节点即为其前驱

③如果能找到 p 的父节点，且p是右孩子，其左兄弟非空，则p的前驱为其左兄弟子树最后一个被先序遍历的结点

④如果p是根节点，则p没有先序前驱

后序线索  
二叉树

后继

①如果能找到 p 的父节点，且p是右孩子，p的父节点即为其后继

②如果能找到 p 的父节点，且p是左孩子，其右兄弟为空，p的父节点即为其后继

③如果能找到 p 的父节点，且p是左孩子，其右兄弟非空，则p的后继为其右兄弟子树中第一个被后序遍历的结点

④如果p是根节点，则p没有后序后继

前驱

①若p有右孩子，则后序前驱为右孩子

②若p没有右孩子，则后序前驱为左孩子



## 知识回顾与重要考点

	中序线索二叉树	先序线索二叉树	后序线索二叉树
找前驱	✓	✗	✓
找后继	✓	✓	✗

除非用三叉链表，  
或者用土办法从根  
开始遍历寻找

线索二叉树高频考点

线索化

! 手算

代码

! 找前驱、找后继